ELSEVIER

Journal of Computational and Applied Mathematics 145 (2002) 49–70

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Efficient algebraic solution of reaction–diffusion systems for the cardiac excitation process

Micol Pennacchio *, Valeria Simoncini

Istituto di Analisi Numerica - CNR via Ferrata 1, 27100 Pavia, Italy

Received 2 September 2000; received in revised form 20 August 2001

Abstract

In this paper, we deal with the problem of solving the large algebraic linear system arising in the numerical solution of a reaction–diffusion (R–D) system associated with myocardial excitation process modeling. We show that an ad hoc preconditioning technique can be devised so as to efficiently and simultaneously handle the differential equations of the R–D system, with no additional memory requirements.

Two different formulations are commonly considered for the theoretical and numerical analyses, respectively. We observe that the formulation employed for the theoretical analysis of the problem actually yields the best numerical performance, when compared with the usual numerical scheme. © 2001 Elsevier Science B.V. All rights reserved.

MSC: 65F10; 65F15; 35K57; 35K65

Keywords: Reaction–diffusion system; Iterative methods; Preconditioning

1. Introduction

Three dimensional simulations of the heart's electrical activity are a computationally intensive task, even on a small block of cardiac tissue. This is due to the fact that dealing with the reaction–diffusion (R–D) system that models the myocardial excitation process is very costly both in terms of time and computer memory. Indeed, the first phase of the excitation process, known as depolarization, is characterized by a steep propagating layer having a thickness of about 1 mm and spreading through the myocardium with an upstroke lasting about 1 ms. Therefore the numerical solution of the problem requires small space and time steps (of the order of 0.1 mm and 0.05 ms, respectively) so that numerical simulations restricted to a 3D block of few cm dimension can be handled in practice [9,20]. Alternatively, large scale simulations may be obtained using an eikonal approach;

* Corresponding author.

E-mail addresses: micol@ian.pv.cnr.it (M. Pennacchio), val@ian.pv.cnr.it (V. Simoncini).

PII: S0377-0427(01)00535-0

this mathematical model, however, is only valid for the first phase of the excitation process, i.e., for the depolarization but not for the subsequent phases [9–11,23]. As a consequence, to correctly simulate all phases of the excitation process, we are faced with the problem of efficiently handling the full reaction–diffusion system. At each time step, this entails the solution of a large linear algebraic system, whose computational cost is dominant with respect to the other parts of the computation. Therefore, efficient and low memory consuming iterative solvers are fundamental tools to make the entire numerical formulation feasible.

The R–D system is characterized by the coupling of the intra- and extracellular potentials $\bar{u}, u$ through a degenerate temporal structure, for which existence and uniqueness results can be found in [12]. However, most numerical simulations in the literature were obtained by pairing a parabolic and an elliptic equation in u and v, respectively, where $v = \bar{u} - u$ is the transmembrane potential [9,20,21,24,30]. This formulation leads to a nested (in time) scheme for the parabolic equation, in which an algebraic system associated with the elliptic problem is solved at each iteration. This approach is usually preferred because it explicitly isolates the solution of the elliptic problem from the rest of the computation. In algebraic terms, the most natural generalization of the continuous formulation consists of using a block Gauss–Seidel iterative method, where the blocks involved correspond to the discretization of the elliptic and parabolic equations; this has been exploited for instance in [9].

It is important to notice, however, that the discretization of both formulations $(\bar{u}, u)$ and (u, v) leads to a linear system with a block coefficient matrix, which can be solved by means of different iterative methods, other than block Gauss–Seidel, whose choice was somehow misguided by the elliptic–parabolic structure of the (u, v) formulation.

The aim of this paper is twofold: (i) we show that dealing with the large block system may be extremely advantageous if the properties of the coefficient matrix are taken into account; (ii) we show that the $(\bar{u}, u)$ formulation yields a convenient matrix structure, which leads to a more efficient system solution than that obtained with the commonly used (u, v) formulation, in terms of computer time.

We show that the conjugate gradient method (CG) is very efficient on this problem when ad hoc preconditioning techniques are used. More precisely, we analyze a block SSOR preconditioner that substantially reduces the time spent to solve the linear system with no additional memory requirements. A convenient permutation of the matrix entries allows to fully exploit the inherent properties of the reaction–diffusion problem. Comparisons with state-of-the-art techniques and with iterative methods previously used are presented, showing the efficiency, in terms of time and memory, of the new approach and the easiness of its implementation. As a side result, we also estimate the rate of convergence of the classical block Gauss–Seidel method on the two formulations.

Although our numerical simulations aim at modeling blocks of cardiac tissue of limited dimension, experimental results are very promising, and may be considered as a first step towards large scale simulations, that is for large portions of myocardium. It should be remarked, however, that in order to make 3-D large scale numerical simulations feasible, some form of spatial and/or time adaptivity need be considered; first attempts in this direction can be found in [34,31,35].

Recently, alternative approaches based on multigrid [24] have been proposed that rely on the properties of the associated differential operators. However, in the case an adaptive methodology is considered, highly unstructured grids are constructed, in order to match the very complex topology of the cardiac excitation wavefront. In this context and in the presence of nonlinear terms, multigrid

type schemes may present difficulties when transferring solutions between grids. These problems are completely avoided in fully algebraic approaches.

The paper is organized as follows. In Section 2, we recall the $(\bar{u}, u)$ formulation and describe some properties of the functions and bilinear forms associated with the problem. In Section 3, we briefly describe the numerical discretization of the continuous problem, while in Section 3.1, we focus on the algebraic aspects of the resulting discrete problem. More precise results on the spectral properties of the matrices are also stated. Section 4 is devoted to the analysis of the block Gauss–Seidel method for the $(u, \bar{u})$ formulation, whereas the new preconditioning approach is fully described in Sections 5 and 5.1. The (v, u) formulation is recalled in Section 6 and the associated block Gauss–Seidel method is analyzed in Section 6.1. Finally, numerical experiments are presented in Section 7 and our conclusions are summarized in Section 8.

The following notation will be used: bold face indicates real vectors and "T" indicates vector and matrix real transposition. The notation $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ denotes a $2n$ vector given by the two vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ stacked one after the other. Matlab [25] notation is used whenever possible. The vector $\mathbf{e}$ denotes the vector of all ones and its dimension is clear from the context. The notation (A, B) indicates the matrix pencil associated with the generalized eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$.

2. Mathematical model

The excitation process in the myocardium is a complex phenomenon characterized by rapid ionic fluxes through the cellular membrane separating the intracellular and the interstitial fluid in the myocardium. A well known macroscopic representation of the cardiac tissue is given by the *anisotropic bidomain*: the myocardium is seen as two interpenetrating anisotropic continua, intracellular (i) and extracellular (e), connected everywhere by the distributed cellular membrane [19].

The governing equations take into account the currents in and between each of these spaces, as well as the potentials generated by these currents. The macroscopic bidomain model for the description of the intra- and extracellular potential yields the following nonlinear degenerate parabolic system:
given $I_{\text{app}} : \Omega \times]0, T[\to \mathbf{R}$ *and* $v_0 : \Omega \to \mathbf{R}$,
find $\bar{u}, u : \Omega \times]0, T[\to \mathbf{R}$ *and* $v = \bar{u} - u$ *such that*:

$$
\begin{aligned}
& c_{\text{m}} \partial_t v - \operatorname{div} M_{\text{i}} \nabla \bar{u} + I(v) = I_{\text{app}} \quad \text{in } \Omega \times]0, T[, \\
& c_{\text{m}} \partial_t v + \operatorname{div} M_{\text{e}} \nabla u + I(v) = I_{\text{app}} \quad \text{in } \Omega \times]0, T[, \\
& \mathbf{n}^{\text{T}} M_{\text{i}} \nabla \bar{u} = 0 \quad \text{on } \Gamma \times]0, T[, \\
& \mathbf{n}^{\text{T}} M_{\text{e}} \nabla u = 0 \quad \text{on } \Gamma \times]0, T[, \\
& v(\mathbf{x}, 0) = v_0(\mathbf{x}) \quad \text{in } \Omega,
\end{aligned}
\tag{2.1}
$$

where $\Omega \subset \mathbf{R}^3$ models the heart tissue, $\Gamma = \partial\Omega, \bar{u}(\mathbf{x}, t), u(\mathbf{x}, t)$ are the intra- and extracellular potential, $v(\mathbf{x}, t) = \bar{u}(\mathbf{x}, t) - u(\mathbf{x}, t)$ is the transmembrane potential and $\mathbf{n}$ denotes the outward unit normal to the boundary Γ. The anisotropic properties of the media are modeled by the intra- and extracellular conductivity tensors $M_{\text{i}} = M_{\text{i}}(\mathbf{x})$ and $M_{\text{e}} = M_{\text{e}}(\mathbf{x})$; $c_{\text{m}} = \chi C_{\text{m}}$ represents the surface capacitance of the membrane and χ the ratio of the membrane area per unit tissue volume. The homogeneous Neumann boundary conditions reflect the fact that tissue is surrounded by an insulator.

The function $I(v)$ is the transmembrane ionic current which is assumed for simplicity to depend only on v. There are additional ordinary differential equations governing the evolution of v, the analysis of which is beyond the scope of this work. Moreover, we are interested only in the depolarization phase since it is the most expensive step in terms of time and computer memory requirements. Hence, we consider the current $I(v) = \chi I_{\text{ion}}(v)$ with $I_{\text{ion}}(v) = Gv(1 - v/v_{\text{th}})(1 - v/v_{\text{p}})$ where G is the maximum membrane conductance per unit area and v_{th}, v_{p} are the threshold and plateau values of v (see [19] for more details on ionic currents). Finally, the term $I_{\text{app}} = \chi^{\text{iapp}}(\mathbf{x}, t)$ models an applied current used to initiate the process.

We recall that the conductivity tensors can be written as

$$M_s = \sigma_t^s I + (\sigma_l^s - \sigma_t^s)\mathbf{a}\mathbf{a}^{\mathrm{T}} \quad s = \mathrm{i}, \mathrm{e}, \tag{2.2}$$

where $\mathbf{a} = \mathbf{a}(\mathbf{x})$ is the unit tangent to the cardiac fiber at a point $\mathbf{x} \in \Omega$ and σ_l^s, σ_t^s for $s = \mathrm{i}, \mathrm{e}$ are the conductivity coefficients, along and across fiber, in the (i) and (e) media, assumed constant with

$$\sigma_l^s > \sigma_t^s > 0 \quad \text{for } s = \mathrm{i}, \mathrm{e}. \tag{2.3}$$

From (2.2) it is easy to verify that $M_{\mathrm{i}}, M_{\mathrm{e}}$ are symmetric positive definite matrices with two different distinct eigenvalues $\sigma_{l,t}^{\mathrm{i}}$ and $\sigma_{l,t}^{\mathrm{e}}$ respectively, both positive; multiplicity of σ_l^s and σ_t^s for $s = \mathrm{i}, \mathrm{e}$ is 1 and 2, respectively. More specifically we have

$$\sigma_t^s |\boldsymbol{\xi}|^2 \leqslant \boldsymbol{\xi}^{\mathrm{T}} M_s(\mathbf{x}) \boldsymbol{\xi} \leqslant \sigma_l^s |\boldsymbol{\xi}|^2 \quad \forall \boldsymbol{\xi} \in \mathbf{R}^3, \ \ \mathbf{x} \in \Omega, \ \ s = \mathrm{i}, \mathrm{e}. \tag{2.4}$$

The reaction–diffusion system (2.1) is characterized by the coupling of $\bar{u}$ and u through a degenerate temporal structure. In [12] existence and uniqueness results are shown for a more general ionic current $I(v)$, i.e., the solution $(\bar{u}, u) \in L^2(0, T; H^1(\Omega))$ is uniquely determined up to a family of additive constants $c(t)$. If $I(v)$ is a cubic polynomial then a strong solution $(\bar{u}, u)$ exists and $(\bar{u}, u) \in L^2(0, T; H^2(\Omega))$ (see [12]). Problem (2.1) admits the following variational formulation:

find $\bar{u}, u :]0, T[\to H^1(\Omega)$ such that:

$$\begin{aligned} &c_{\mathrm{m}} \frac{\mathrm{d}}{\mathrm{d}t}(v(t), \varphi) + a_{\mathrm{i}}(\bar{u}(t), \varphi) + (I_{\text{ion}}(v(t)), \varphi) = (I_{\text{app}}(t), \varphi) \quad \forall \varphi \in V, \\ &c_{\mathrm{m}} \frac{\mathrm{d}}{\mathrm{d}t}(v(t), \varphi) - a_{\mathrm{e}}(u(t), \varphi) + (I_{\text{ion}}(v(t)), \varphi) = (I_{\text{app}}(t), \varphi) \quad \forall \varphi \in V, \end{aligned} \tag{2.5}$$

where $v(t) = \bar{u}(t) - u(t)$, $V = H^1(\Omega)$, $(\psi, \phi) = \int_\Omega \psi \ \phi \, \mathrm{d}x \ \forall \psi, \phi \in L^2(\Omega)$ and $a_s(\cdot, \cdot) : V \times V \to \mathbf{R}$ $s = \mathrm{i}, \mathrm{e}$ are the bilinear symmetric continuous forms defined as

$$a_s(\psi, \phi) \int_\Omega (\nabla \psi)^{\mathrm{T}} M_s \nabla \phi \, \mathrm{d}x \quad \forall \psi, \phi \in V \ \ s = \mathrm{i}, \mathrm{e}. \tag{2.6}$$

Let P_0 be the set of constant functions, $Q = \{w \in H^1(\Omega) : \int_\Omega w \, \mathrm{d}x = 0\}$ and let $||\cdot||_1$ be the usual H^1-norm. Then the forms $a_s(\cdot, \cdot)$ for $s = \mathrm{i}, \mathrm{e}$ satisfy

$$a_s(w, w) \leqslant \gamma ||w||_1^2 \quad \forall w \in H^1(\Omega) \ \ s = \mathrm{i}, \mathrm{e}, \tag{2.7}$$

$$a_s(w, w) \geqslant \alpha ||w||_1^2 \quad \forall w \in Q \ \ s = \mathrm{i}, \mathrm{e} \tag{2.8}$$

with $\gamma = \min(\sigma_l^e, \sigma_l^i)$, $\alpha = \min(\sigma_t^s/2, \sigma_t^s/2c)$ and c the Poincaré constant. Moreover, if $u \in P_0$ then $a_s(u, w) = 0$, $\forall w \in H^1(\Omega)$.

3. Numerical approximation

A first step towards the approximation of the solution of (2.5) entails only the space discretization. We model a block of myocardium as a parallelepipedal slab of dimension (l_1, l_2, l_3) with edges parallel to the x-, y-, z-axis. A uniform mesh $\mathscr{T}_h$ on Ω, made up of parallelepipedal elements with edges (h_1, h_2, h_3), where $h_i = l_i/n_i$, $i = 1, 2, 3$, is considered.

The variational formulation (2.5) leads to a semi-discrete problem by approximating the space V with a finite dimensional space V_h. In this work, V_h is the space of continuous functions on Ω whose restriction to each parallelepipedal element are trilinear polynomials [32].

The semi-discrete approximate problem reads as follows:

for each $t \in]0, T[$ find $(\bar{u}_h(t), u_h(t)) \in V_h \times V_h$ such that:

$$\begin{aligned} &c_m \frac{d}{dt}(v_h(t), \phi_h) + a_i(\bar{u}_h(t), \phi_h) + (I(v_h(t)), \phi_h) = (I_{app}(t), \phi_h) \quad \forall \phi_h \in V_h, \\ &c_m \frac{d}{dt}(v_h(t), \phi_h) - a_e(u_h(t), \phi_h) + (I(v_h(t)), \phi_h) = (I_{app}(t), \phi_h) \quad \forall \phi_h \in V_h, \\ &v_h(0) = v_{0,h}, \end{aligned} \tag{3.1}$$

where $v_{0,h}$ is a FEM interpolation of v_0. Looking for an approximate solution $\bar{u}_h(t) = \sum_{j=1}^n \bar{u}_j(t)\phi_j$, $u_h(t) = \sum_{j=1}^n u_j(t)\phi_j$, where $\{\phi_j\}$ is a basis of V_h and $n = \prod_{i=1}^3 (n_i + 1)$, we derive a first order system of nonlinear ordinary differential equations in $\boldsymbol{\xi}(t) = (\bar{\mathbf{u}}(t), \mathbf{u}(t))$ with $\bar{\mathbf{u}}(t) = [\bar{u}_1(t), \ldots, \bar{u}_n(t)]^T$, $\mathbf{u}(t) = [u_1(t), \ldots, u_n(t)]^T$ vectors of nodal values of $\bar{u}(\mathbf{x}, t)$ and $u(\mathbf{x}, t)$. Thus, we approximate Problem (3.1) as follows:

find $(\bar{\mathbf{u}}(t), \mathbf{u}(t))$ solution of

$$\begin{aligned} &c_m C \frac{d\mathbf{v}}{dt} + \mathbf{i}(\mathbf{v}) + A_i \bar{\mathbf{u}} = \mathbf{i}_a(t), \\ &c_m C \frac{d\mathbf{v}}{dt} + \mathbf{i}(\mathbf{v}) - A_e \bar{\mathbf{u}} = \mathbf{i}_a(t), \end{aligned} \tag{3.2}$$

where $\mathbf{v} = \bar{\mathbf{u}} - \mathbf{u}$ and

$$C = \left\{ c_{i,j} = \sum_{K \in \mathscr{T}_h} \int_K \phi_i \phi_j \, dx; \ j, i = 1, \ldots, n \right\}, \tag{3.3}$$

$$A_s = \left\{ a_{i,j} = \sum_{K \in \mathscr{T}_h} \int_K (\nabla \phi_i)^T M_s \nabla \phi_j \, dx; \ i, j = 1, \ldots, n \right\}, \quad s = i, e, \tag{3.4}$$

$$\mathbf{i}(\mathbf{v}) = C\mathbf{I}(\mathbf{v}) \quad \text{with } \mathbf{I}(\mathbf{v}) = (I(v_1(t)), \ldots, I(v_n(t)))^{\mathrm{T}}, \tag{3.5}$$

$$\mathbf{i}_a(t) = \left\{ \int_\Omega I_{\mathrm{app}}(t)\phi_j \, \mathrm{d}x; \ j = 1, \ldots, n \right\}. \tag{3.6}$$

The nonlinear term $\mathbf{i}(\mathbf{v})$ was computed using the product approximation

$$\mathbf{i}_k(\mathbf{v}) = \int_\Omega I\left(\sum_{i=1}^{n} v_i(t)\phi_i(\mathbf{x})\right)\phi_k(\mathbf{x})\,\mathrm{d}x \simeq \int_\Omega \left(\sum_{i=1}^{n} I(v_i(t))\phi_i(\mathbf{x})\right)\phi_k(\mathbf{x})\,\mathrm{d}x$$

$$= \sum_{i=1}^{n} I(v_i(t)) \int_\Omega \phi_i\phi_k \,\mathrm{d}x = (C[I(v_1(t)), \ldots, I(v_n(t))]^{\mathrm{T}})_k.$$

The entries of the matrices are obtained by using a three-dimensional trapezoidal quadrature rule, so that the mass matrix C reduces to a (diagonal) lumped matrix.

Properties of the continuous problem reflect into analogous properties of the matrices A_s for $s = \mathrm{i}, \mathrm{e}$. More specifically using (2.2), (2.3) and standard finite element arguments, we can assert that A_s, $s = \mathrm{i}, \mathrm{e}$ are symmetric and positive semidefinite with $A_s\mathbf{e} = 0$; in particular, $a_{k,k}$ for $k = 1, n$ are positive and $a_{k,k} = -\sum_{j \neq k} a_{j,k}$. Note that the off-diagonal entries of A_s may assume either positive or negative sign.

3.1. Algebraic form

The numerical discretization of parabolic partial differential equations is a well studies problem, see for instance [1]. The methods primarily used for the time discretization of (3.1) are: fully explicit (forward Euler), fully implicit (backward Euler), mixed explicit–implicit (semi-implicit) and Crank–Nicholson.

The fully explicit method is easy to implement but excessively small time steps must be considered. A fully implicit scheme, however, requires the implicit treatment of the nonlinear reaction term $I(\mathbf{v})$, hence the solution of a large nonlinear system of equations at each time step [30,21,22].

An interesting compromise is given by semi-implicit methods: they are stable for large time steps than explicit schemes, but require the solution of a linear system of equations every step. Typically an implicit scheme is chosen for the diffusion term whereas an explicit scheme is applied to the reaction term.

In this work, we mostly focus on the efficient solution of the associated algebraic system. If a semi-implicit scheme is used to discretize (3.1), then the following general algebraic system can be obtained:

$$\mathscr{A}\boldsymbol{\xi}^{k+1} = \mathbf{b} \quad \text{with } \mathscr{A} = \begin{bmatrix} C_t + \theta A_{\mathrm{i}} & -C_t \\ -C_t & C_t + \theta A_{\mathrm{e}} \end{bmatrix} \tag{3.7}$$

with $\mathbf{b} = [C_t\mathbf{v}^k - \mathbf{i}(\mathbf{v}^k) + \mathbf{i}_a - \beta A_{\mathrm{i}}\bar{\mathbf{u}}^k; -C_t\mathbf{v}^k + \mathbf{i}(\mathbf{v}^k) - \mathbf{i}_a - \beta A_{\mathrm{e}}\mathbf{u}^k]$, $C_t = (c_{\mathrm{m}}/\tau)C$ diagonal with positive diagonal entries, τ the time step, $\mathbf{v}^k = \bar{\mathbf{u}}^k - \mathbf{u}^k$ and $\boldsymbol{\xi}^{k+1} = [\bar{\mathbf{u}}^{k+1}; \mathbf{u}^{k+1}]$. Here the constants θ, β depend on the specific semi-implicit scheme used, however, the dependence of $\mathscr{A}$ on θ does not significantly influence the main properties of the system. The values of θ for the most commonly used methods are reported in Table 1. In the definition of the right-hand side $\mathbf{b}$, $\beta = 0$ and $1/2$ are associated with the forward–backward Euler and the Crank–Nicholson schemes, respectively.

Table 1
System parameter θ for commonly used discretization methods

Method	θ
f.Euler + b.Euler[a]	1
Crank–Nicholson	1/2
Predictor–Corrector [9]	1/2

[a]Forward Euler for the nonlinear term, backward Euler for the diffusive part.

In the following, we shall consider the case $\theta = 1$ and $\beta = 0$, while our algebraic results can be trivially stated in terms of general θ.

Matrix $\mathscr{A}$ is positive semidefinite, indeed, for $\mathbf{0} \neq \boldsymbol{\xi} = [\bar{\mathbf{u}}; \mathbf{u}] \in \mathbf{R}^{2n}$, we have

$$\begin{aligned}\boldsymbol{\xi}^{\mathrm{T}} \mathscr{A} \boldsymbol{\xi} &= \bar{\mathbf{u}}^{\mathrm{T}} C_t \bar{\mathbf{u}} - 2\mathbf{u}^{\mathrm{T}} C_t \bar{\mathbf{u}} + \mathbf{u}^{\mathrm{T}} C_t \mathbf{u} + \bar{\mathbf{u}}^{\mathrm{T}} A_{\mathrm{i}} \bar{\mathbf{u}} + \mathbf{u} A_{\mathrm{e}} \mathbf{u} \\ &= (\bar{\mathbf{u}} - \mathbf{u})^{\mathrm{T}} C_t (\bar{\mathbf{u}} - \mathbf{u}) + \bar{\mathbf{u}}^{\mathrm{T}} A_{\mathrm{i}} \bar{\mathbf{u}} + \mathbf{u}^{\mathrm{T}} A_{\mathrm{e}} \mathbf{u} \geqslant 0,\end{aligned}$$

since A_s, $s = \mathrm{i}, \mathrm{e}$ are positive semidefinite and C_t is positive definite. Moreover, $\mathscr{A}[\mathbf{e}; \mathbf{e}] = \mathbf{0}$ and the system is consistent, in that $\mathbf{b}$ has zero mean, that is $\mathbf{e}^{\mathrm{T}} \mathbf{b} = \mathbf{0}$.

We next give bounds for the smallest nonzero eigenvalue of the matrices $A_{\mathrm{i}}, A_{\mathrm{e}}$ in terms of the mesh parameter h, from which corresponding spectral bounds for $\mathscr{A}$ will also follow. Denoting by $\lambda_{\min}(A_s)$ the smallest nonzero eigenvalue of A_s, $s = \mathrm{i}, \mathrm{e}$ and recalling that the space dimension is three, as in the nonsingular case it can be easily verified that

$$ch^3 \leqslant \lambda_{\min}(A_s) \leqslant dh^3$$

with c, d constants [4]. Scaling A_s by its diagonal leads to a less dramatic situation. Indeed, using standard finite element arguments, it can be shown that $\lambda_{\min}(A_s, D_{A_s}) = \mathscr{O}(h^2)$ as $h \to 0$, for $s = \mathrm{i}, \mathrm{e}$, where D_{A_s} indicates the matrix of the diagonal entries of A_s. The bounds above will be used in Section 5.1 in the convergence analysis of the block SSOR preconditioner.

4. Classical algebraic approaches

The linear system to be solved using the formulation of the previous section is

$$\mathscr{A}\boldsymbol{\xi} \equiv \begin{bmatrix} C_t + A_{\mathrm{i}} & -C_t \\ -C_t & C_t + A_{\mathrm{e}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \tag{4.1}$$

Following the classical methodology originally used in the (v, u) formulation, the system above could be solved by means of the block version of the Gauss–Seidel method [33,26]. This iterative scheme exploits the splitting

$$\begin{bmatrix} C_t + A_{\mathrm{i}} & -C_t \\ -C_t & C_t + A_{\mathrm{e}} \end{bmatrix} = \begin{bmatrix} C_t + A_{\mathrm{i}} & 0 \\ -C_t & C_t + A_{\mathrm{e}} \end{bmatrix} - \begin{bmatrix} 0 & C_t \\ 0 & 0 \end{bmatrix}, \tag{4.2}$$

where the first addend is nonsingular. We shall see, however, that this splitting leads to very slow convergence of the iterative scheme, which discouraged researches from pursuing the use of the $(\bar{u}, u)$ formulation for numerical purposes (cf. the discussion in Section 1). In Section 5, we propose

an alternative approach to efficiently handle system (4.1). The block Gauss–Seidel iteration matrix is

$$B_{\mathrm{GS}} = \begin{bmatrix} C_t + A_{\mathrm{i}} & 0 \\ -C_t & C_t + A_{\mathrm{e}} \end{bmatrix}^{-1} \begin{bmatrix} 0 & C_t \\ 0 & 0 \end{bmatrix} \tag{4.3}$$

and systems with the symmetric positive definite coefficient matrices $C_t + A_{\mathrm{i}}$ and $C_t + A_{\mathrm{e}}$ need be solved at each iteration; we refer to standard numerical linear algebra texts (cf. e.g. [15,33,26]) for a description of the method.

The rate of convergence of the iterative scheme depends on the spectral properties of B_{GS}. Since the original matrix is singular, it can be easily shown that the matrix B_{GS} has one unit eigenvalue corresponding to the eigenvector $[\mathbf{e}; \mathbf{e}]$. Nevertheless, the following property holds [13,7]. We recall that the index of an eigenvalue is the dimension of its largest Jordan block [7].

Theorem 4.1. *Let $\mathscr{A} = \mathscr{M} - \mathscr{N}$ be a splitting of the singular matrix $\mathscr{A}$ with $\mathscr{M}$ nonsingular and let λ be an eigenvalue of $\mathscr{M}^{-1}\mathscr{N}$. Then the iterative method with iteration matrix $\mathscr{M}^{-1}\mathscr{N}$ converges if and only if either $|\lambda| < 1$, or $|\lambda| = 1$ with index 1.*

The unit eigenvalue of the iteration matrix B_{GS} has multiplicity (and therefore index) equal to one, therefore the corresponding block Gauss–Seidel iteration converges. In this case, the rate of convergence depends on the nonunit eigenvalue of $\mathscr{M}^{-1}\mathscr{N}$ closest to one in modulo [7]. The proof of the following result is postponed to the appendix.

Proposition 4.2. *Let λ be an eigenvalue of B_{GS} as defined in* (4.3). *Then either $\lambda = 1$ with multiplicity one or*

$$0 \leqslant \lambda \leqslant \frac{1}{(1 + \lambda_{\min}(A_{\mathrm{e}}, C_t))(1 + \lambda_{\min}(A_{\mathrm{i}}, C_t))} < 1,$$

where $\lambda_{\min}(A_s, C_t)$, $s = \mathrm{i}, \mathrm{e}$ indicates the smallest nonzero eigenvalue of the pencil (A_s, C_t).

The upper inequality for λ indicates that if $\lambda_{\min}(A_{\mathrm{i}}, C_t), \lambda_{\min}(A_{\mathrm{e}}, C_t)$ are much smaller than one, then the upper bound for the nonunit eigenvalues is in fact very close to 1. We have found numerically that the bound is quite sharp and that is indeed close to 1, implying slow convergence of the block Gauss–Seidel method on $\mathscr{A}$.

5. Preconditioned conjugate gradients

Competitive alternatives to classical iterative methods are Krylov subspace methods [33]. Since our coefficient matrix is positive semidefinite and the corresponding system (4.1) consistent, the conjugate gradient method is applicable [15] and its rate of convergence is governed by the ratio between the largest and nonzero smallest eigenvalues of $\mathscr{A}$. In order to enhance convergence, however, preconditioning of the system is usually carried out. This amounts to determining a matrix $\mathscr{P}$ such that the preconditioned system $\mathscr{P}^{-1}\mathscr{A}d = \mathscr{P}^{-1}b$ is easier to solve than the original system; the matrix

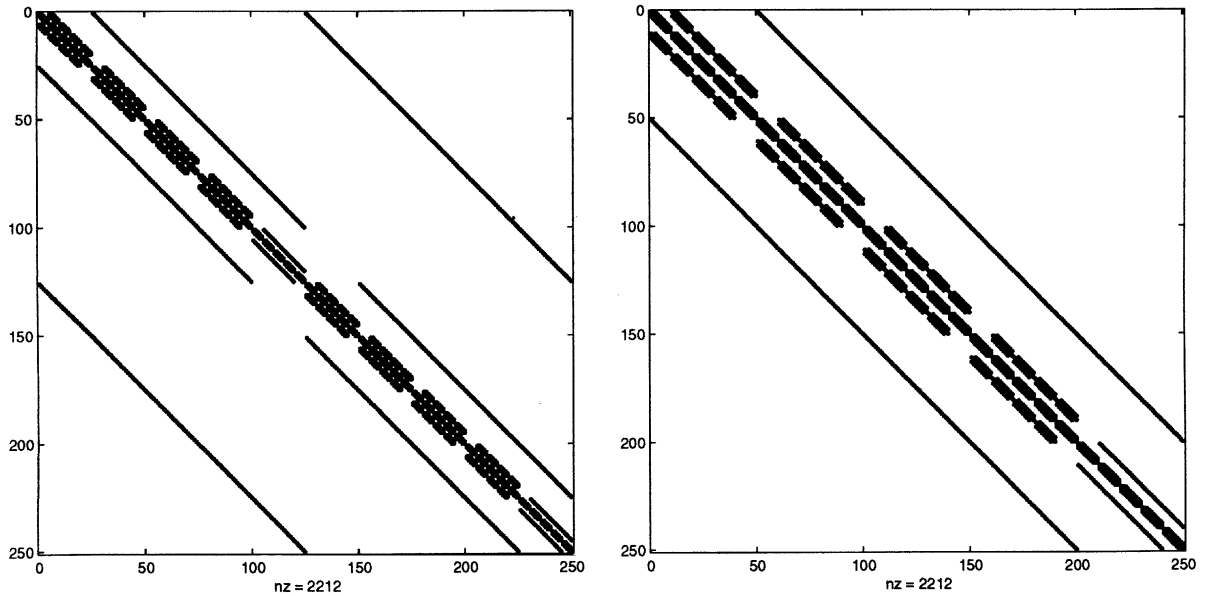

Fig. 1. Sparsity pattern before (Left) and after (Right) reordering for a typical problem matrix.

$\mathscr{P}$ is chosen so that solving systems with $\mathscr{P}$ is cheap. We refer to [26,33] for a comprehensive treatment of standard preconditioning techniques.

In order to fully exploit the structure of the system matrix $\mathscr{A}$, we first reorder the matrix entries so that the diagonal elements of the $(1,2)$ block $-C_t$ move to the diagonal of $\mathscr{A}$ closest to the main diagonal (the same holds for the lower $(2,1)$ block): the permutation vector is simply given by

$$[1, n+1, 2, n+2, \ldots, n-1, 2n-1, n, 2n].$$

This strategy aims at exploiting the fact that the nonzero entries of C_t seem, at least numerically, to be dominant with respect to the nondiagonal entries of $A_\mathrm{e}, A_\mathrm{i}$.

After permutation, matrix $\mathscr{A}$ can be written as

$$\tilde{\mathscr{A}} = D + L + L^{\mathrm{T}},$$

where L is strictly block lower triangular, D is a block diagonal matrix with 2×2 blocks. The pattern of the original and permuted matrices is given in Fig. 1, left and right plots, respectively. Each 2×2 block of the matrix D is of the form

$$\begin{bmatrix} a_{k,k}^{(s)} + c_{k,k} & -c_{k,k} \\ -c_{k,k} & a_{k+1,k+1}^{(s)} + c_{k+1,k+1} \end{bmatrix} \quad s = \mathrm{i}, \mathrm{e},$$

where we have denoted $A_s = (a_{k,j}^{(s)})$, $C_t = (c_{k,j})$.

We then consider the block symmetric SOR (block SSOR) preconditioner, that is [3,33]

$$\mathscr{P}_\omega = (\omega L + D) D^{-1} (\omega L^{\mathrm{T}} + D)$$

with $\mathscr{P}_\omega$ symmetric positive definite, and we solve the preconditioned system

$$\mathscr{P}_\omega^{-1} \tilde{\mathscr{A}} x = \mathscr{P}_\omega^{-1} b.$$

The parameter ω plays the same role as in the classical block SSOR method. The advantages of this preconditioner are well known [3,33,26]: no additional memory is needed for storing the preconditioner and no computation is required to generate the factors. These features make the approach particularly attractive on our 3D problem of large dimension. Block SSOR preconditioning has been extensively used, for instance in connection with hierarchical FE approximation [16,5] and in high-performance computer architecture [18,2,36,17]. Although in general incomplete factorization preconditioners may be more efficient, SSOR-type preconditioning strategies are very advantageous in case, e.g., of particular problem environments, where memory and architecture constraints require to only employ the coefficient matrix entries.

It is known that the application of the preconditional matrix $\mathscr{P}_\omega^{-1}\tilde{\mathscr{A}}$ to a vector can be implemented so that the multiplication with $\tilde{\mathscr{A}}$ can be avoided, therefore reducing the computational effort. This is done as follows by the so-called Eisenstat's trick [14]. We apply the conjugate gradients algorithm to the linear system

$$\hat{\mathscr{M}}x = (D+\omega L)^{-1}\mathbf{b}$$

with $\hat{\mathscr{M}} = (D+\omega L)^{-1}(D+L+L^{\mathrm{T}})(D+\omega L^{\mathrm{T}})^{-1}$; see [33] for the case $\omega = 1$. Therefore, we can write

$$\begin{aligned}\hat{\mathscr{M}} &= (D+\omega L)^{-1}\left[D+\frac{2}{\omega}D+\frac{1}{\omega}(D+\omega L)+\frac{1}{\omega}(D+\omega L^{\mathrm{T}})\right](D+\omega L^{\mathrm{T}})^{-1}\\ &= (D+\omega L)^{-1}\left(1-\frac{2}{\omega}\right)D(D+\omega L^{\mathrm{T}})^{-1}+\frac{1}{\omega}(D+\omega L^{\mathrm{T}})^{-1}+\frac{1}{\omega}(D+\omega L)^{-1}.\end{aligned}$$

A matrix–vector multiplication can be thus carried out as

$$\mathbf{d} = \hat{\mathscr{M}}\mathbf{v} = (D+\omega L)^{-1}\left[\left(1-\frac{2}{\omega}\right)D(D+\omega L^{\mathrm{T}})^{-1}\mathbf{v}+\frac{1}{\omega}\mathbf{v}\right]+\frac{1}{\omega}(D+\omega L^{\mathrm{T}})^{-1}\mathbf{v}$$

using the following steps:

- $\mathbf{z} = \frac{1}{\omega}(D+\omega L^{\mathrm{T}})^{-1}\mathbf{v}$,
- $\mathbf{d} = (D+\omega L)^{-1}(\frac{1}{\omega}\mathbf{v}+\omega(1-\frac{2}{\omega})D\mathbf{z})$,
- $\mathbf{d} = \mathbf{d}+\mathbf{z}$

so that the cost of multiplying by $\tilde{\mathscr{A}}$ is replaced by the cheaper multiplication with the block diagonal matrix D.

The value of ω is chosen so that the condition number of $\mathscr{P}_\omega^{-1}\tilde{\mathscr{A}}$ is closest to one. Although it is usually reported that the selection of ω in the preconditioning context is not as crucial as in the SOR method, we have experimentally noticed that the performance of preconditioned CG was indeed sensitive to the choice of the parameter ω. In the next section, we thus analyze the dependence of the convergence of the method on ω, while in the experiments section we report results for $\omega = 1$ and for the (experimentally) optimal value of ω.

Numerical experiments with an incomplete Cholesky factorization as preconditioner have also been carried out and reported in Section 7. Note that no substantial differences were observed between the application of this preconditioner on $\mathscr{A}$ and on the premuted matrix $\tilde{\mathscr{A}}$. Inspection of the sparsity pattern of the preconditioner, showed that this seems to be due to the capability of the used code

on this problem to detect the dominant portions of the coefficient matrix, regardless of the entries ordering.

We would also like to mention that, though not performed here, incomplete preconditioners with no fill-in may be written in the form $P=(\tilde{D}+L)\tilde{D}^{-1}(\tilde{D}+L^{\mathrm{T}})$, where L is the lower part of the permuted coefficient matrix (and thus need not be stored), while $\tilde{D}$ is a (block) diagonal matrix which in general differs from the corresponding section of the coefficient matrix [33]. Additional details on the use of incomplete factorization preconditioning will be given in Section 7.

5.1. Convergence analysis

Results on the convergence when using block as well as pointwise SSOR preconditioning for $\tilde{\mathscr{A}}$ symmetric positive definite can be found in [3], where estimates for the condition number of the preconditioned matrix are given. In our case, the semidefiniteness of $\tilde{\mathscr{A}}$ lead us to consider a simple modification of the classical result.

Proposition 5.1. *Let θ be an eigenvalue of $\mathscr{P}_{\omega}^{-1}\tilde{\mathscr{A}}$ with $\omega\in(0,2)$. Let also $\lambda_{\min}$ be the smallest nonzero eigenvalue of the pencil $(\tilde{\mathscr{A}},D)$ and $\mathbf{f}=[\mathbf{e};\mathbf{e}]$. Then either $\theta=0$ or*

$$\frac{1}{\omega(1+\omega(1/\omega-1/2)^2\lambda_{\min}^{-1}+\omega\gamma)}\leqslant\theta\leqslant\frac{1}{\omega(2-\omega)},\tag{5.1}$$

where

$$\gamma=\max_{\mathbf{0}\neq\mathbf{x}\notin\mathrm{span}\{\mathbf{f}\}}\frac{\mathbf{x}^{\mathrm{T}}(LD^{-1}L^{\mathrm{T}}-1/4D)\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\tilde{\mathscr{A}}\mathbf{x}}.$$

Proof. The preconditioning matrix can be written as

$$\begin{aligned}\mathscr{P}&=\omega^2LD^{-1}L^{\mathrm{T}}+\omega\tilde{\mathscr{A}}+(1-\omega)D\\&=\omega\left(\tilde{\mathscr{A}}+\omega\left(\frac{1}{\omega}-\frac{1}{2}\right)^2D+\omega\left(LD^{-1}L^{\mathrm{T}}-\frac{1}{4}D\right)\right),\end{aligned}\tag{5.2}$$

where we omit the subscript. The upper bound can be simply obtained by using the corresponding bound in [3, Theorem 7.17].

In order to derive the lower bound, we consider the generalized eigenvalue problem $\tilde{\mathscr{A}}\mathbf{x}=\theta\mathscr{P}\mathbf{x}$. We have $0=\tilde{\mathscr{A}}\mathbf{f}=\theta\mathscr{P}\mathbf{f}$ from which $\theta=0$ since $\mathscr{P}$ is nonsingular, so that $(0,\mathbf{f})$ is an eigenpair of the pencil $(\tilde{\mathscr{A}},\mathscr{P})$. All other eigenvectors of $(\tilde{\mathscr{A}},\mathscr{P})$ are orthogonal to $\mathscr{P}\mathbf{f}$. Substituting (5.2) in $\mathbf{x}^{\mathrm{T}}\tilde{\mathscr{A}}\mathbf{x}=\theta\mathbf{x}^{\mathrm{T}}\mathscr{P}\mathbf{x}$, for $\mathbf{x}\notin\mathrm{span}\{\mathbf{f}\}$ we obtain the lower bound (cf. [27]). □

We will show below that $\lambda_{\min}\geqslant ch^2$ as $h\to0$ with c constant, where h is the mesh characteristic dimension. In this case, if $\gamma=\mathcal{O}(1)$ for $h\to0$, by letting $\omega=2/(1+\zeta h)$ for some $\zeta>0$ it follows that the ratio between the largest and nonzero smallest eigenvalues of the preconditioned matrix $\mathscr{P}_{\omega}^{-1}\tilde{\mathscr{A}}$ behaves as $h^{-1},h\to0$, as in the nonsingular case [4]; see [3] for a more detailed discussion; see also [28,29]. A typical behavior is reported in Fig. 2(left), where the symbol "$*$" marks the number of iterations required by the method to achieve convergence for $\omega=1.7$ (cf. Section 7) as

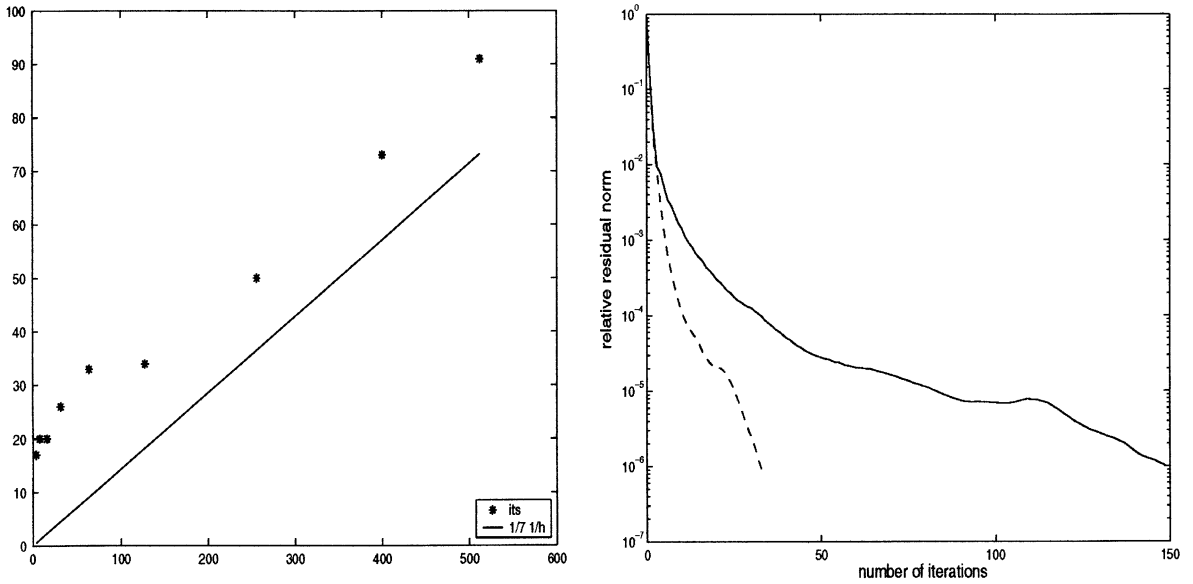

Fig. 2. Left: Number of iterations versus $1/h$ ('$*$') and plot of function $y(1/h) = 1/71/h$('$-$'). Right: Block diagonal preconditioning ($-$) and block symmetric SOR (- -) preconditioning with $\omega = 1.7$.

$1/h$ grows, while the solid line is the plot of the function $y(t) = t/7$ for $t = 1/h$, where h is the longest edge of the parallelepipedal mesh element. Unfortunately, smallest values of h could not be considered because of memory constraints. Nevertheless, it appears from the plot that the iteration growth asymptotically behaves as the linear function $y(1/h)$ on the reference interval.

In order to immediately appreciate the goodness of the SSOR preconditioner, a simple but meaningful comparison is with the less expensive block diagonal (block Jacobi) preconditioner, given by using D as preconditioner. On the model problem in Section 7, block diagonal preconditioning takes 50 s, while block symmetric SOR preconditioning takes 13 s. The convergence history with respect to the number of iterations is reported in Fig. 2 (right), where solid curve corresponds to the block SSOR preconditioned method, and dashed curve to the block diagonally preconditioned scheme.

We next show that the estimates in Section 3.1 for the smallest eigenvalue of the pencils (A_s, D_{A_s}), $s = \mathrm{i}, \mathrm{e}$, also provide a bound, with respect to h, for the smallest nonzero eigenvalue of the pencil $(\tilde{\mathscr{A}}, D)$, as is used in the considerations above.

Proposition 5.2. *Let $\lambda_{\min}$ be the smallest nonzero eigenvalue of the pencil $(\tilde{\mathscr{A}}, D)$. Then $\lambda_{\min} \geqslant ch^2$ for $h \to 0$ with c constant independent of h.*

Proof. The block diagonal matrix D corresponds, in the unpermuted matrix $\mathscr{A}$, to the matrix $D_{\mathscr{A}} + \hat{C}$, where $D_{\mathscr{A}}$ is the diagonal matrix of $\mathscr{A}$ and $\hat{C} = [C_t, -C_t; -C_t, C_t]$. Therefore,

$$\frac{\mathbf{x}^{\mathrm{T}} \tilde{\mathscr{A}} \mathbf{x}}{\mathbf{x}^{\mathrm{T}} D \mathbf{x}} = \frac{\mathbf{y}^{\mathrm{T}} \mathscr{A} \mathbf{y}}{\mathbf{y}^{\mathrm{T}} (D_{\mathscr{A}} + \hat{C}) \mathbf{y}}. \tag{5.3}$$

Let $E = \mathrm{span}\{[\mathbf{e};\mathbf{e}]\}$. Writing $\mathbf{y} = [\mathbf{u};\mathbf{v}]$, we note that $\mathbf{y}^{\mathrm{T}}\hat{C}\mathbf{y} = (\mathbf{u}-\mathbf{v})^{\mathrm{T}}C_t(\mathbf{u}-\mathbf{v})$. Using the spectral bounds on the matrices $D_{A_\mathrm{i}}^{-1}A_\mathrm{i}$ and $D_{A_\mathrm{e}}^{-1}A_\mathrm{e}$ we can write

$$\begin{aligned}
\max_{\mathbf{y}\notin E}\frac{\mathbf{y}^{\mathrm{T}}(D_{\mathscr{A}}+\hat{C})\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathscr{A}\mathbf{y}} &= \max_{[\mathbf{u};\mathbf{v}]\notin E}\frac{\mathbf{u}^{\mathrm{T}}D_{A_\mathrm{i}}\mathbf{u}+\mathbf{v}^{\mathrm{T}}D_{A_\mathrm{e}}\mathbf{v}+(\mathbf{u}-\mathbf{v})^{\mathrm{T}}C_t(\mathbf{u}-\mathbf{v})}{\mathbf{u}^{\mathrm{T}}A_\mathrm{i}\mathbf{u}+(\mathbf{u}-\mathbf{v})^{\mathrm{T}}C_t(\mathbf{u}-\mathbf{v})+\mathbf{v}^{\mathrm{T}}A_\mathrm{e}\mathbf{v}}\\
&\leqslant \max_{[\mathbf{u};\mathbf{v}]\notin E}\left(\frac{\mathbf{u}^{\mathrm{T}}D_{A_\mathrm{i}}\mathbf{u}+\mathbf{v}^{\mathrm{T}}D_{A_\mathrm{e}}\mathbf{v}}{\mathbf{u}^{\mathrm{T}}A_\mathrm{i}\mathbf{u}+(\mathbf{u}-\mathbf{v})^{\mathrm{T}}C_t(\mathbf{u}-\mathbf{v})+\mathbf{v}^{\mathrm{T}}A_\mathrm{e}\mathbf{v}}\right)+1\\
&\leqslant \max_{[\mathbf{u};\mathbf{v}]\notin E}\left(\frac{\mathbf{u}^{\mathrm{T}}D_{A_\mathrm{i}}\mathbf{u}}{\mathbf{u}^{\mathrm{T}}A_\mathrm{i}\mathbf{u}}+\frac{\mathbf{v}^{\mathrm{T}}D_{A_\mathrm{e}}\mathbf{v}}{\mathbf{v}^{\mathrm{T}}A_\mathrm{e}\mathbf{v}}\right)+1\\
&\leqslant c_2h^{-2},\quad h\to 0
\end{aligned}$$

with c_2 constant. Therefore, from [27, Theorem 3.1] and using (5.3),

$$\max_{\mathbf{x}\notin E}\frac{\mathbf{x}^{\mathrm{T}}D\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\tilde{\mathscr{A}}\mathbf{x}} = \max_{\mathbf{x}\perp[\mathbf{e};\mathbf{e}]}\frac{\mathbf{x}^{\mathrm{T}}D\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\tilde{\mathscr{A}}\mathbf{x}} \leqslant c_2h^{-2},$$

so that

$$c_3h^2 \leqslant \min_{\mathbf{x}\perp[\mathbf{e};\mathbf{e}]}\frac{\mathbf{x}^{\mathrm{T}}\tilde{\mathscr{A}}\mathbf{x}}{\mathbf{x}^{\mathrm{T}}D\mathbf{x}} \leqslant \max_{\mathbf{x}\perp D[\mathbf{e};\mathbf{e}]}\frac{\mathbf{x}^{\mathrm{T}}\tilde{\mathscr{A}}\mathbf{x}}{\mathbf{x}^{\mathrm{T}}D\mathbf{x}} \equiv \lambda_{\min}(\tilde{\mathscr{A}},D). \qquad \square$$

6. Previous formulations and methods

System (2.1) may be rewritten into various forms involving different combinations of the variables $\bar{u}, u, v = \bar{u} - u$ and the definition of the bulk conductivity tensor $M = M_\mathrm{i} + M_\mathrm{e}$ [22]. However, most numerical simulations in literature were obtained considering the R–D system written in (v,u) instead of $(\bar{u},u)$, so that problem (2.1) becomes

find $(v(\mathbf{x},t),u(\mathbf{x},t))$, $\mathbf{x}\in H$, $t\in[0,T]$ such that:

$$\begin{aligned}
&c_\mathrm{m}\partial_t v - \mathrm{div}\, M_\mathrm{i}\nabla v + I(v) = \mathrm{div}\, M_\mathrm{i}\nabla u + I_\mathrm{app} \quad \text{in } \Omega\times\,]0,T[,\\
&-\mathrm{div}\, M\nabla u = \mathrm{div}\, M_\mathrm{i}\nabla v \quad \text{in } \Omega\times\,]0,T[,\\
&\mathbf{n}^{\mathrm{T}}M_\mathrm{i}\nabla v = 0 \quad \text{on } \Gamma\times\,]0,T[,\\
&\mathbf{n}^{\mathrm{T}}M\nabla u = 0 \quad \text{on } \Gamma\times\,]0,T[,\\
&v(\mathbf{x},0) = 0 \quad \text{in } \Omega.
\end{aligned} \tag{6.1}$$

The above system differs from problem (2.1) in that the change of variable allows us to replace the second parabolic equation with an elliptic equation, thus loosing the degenerate temporal structure of system (2.1). This approach was usually preferred because the two equations can be solved one after the other.

Finite element discretization yields the approximation
find $(\mathbf{v}(t), \mathbf{u}(t))$ solution of

$$c_{\mathrm{m}} C\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} + \mathbf{i}(\mathbf{v}) + A_{\mathrm{i}}(\mathbf{v}+\mathbf{u}) = \mathbf{i}_a(t), \quad (A_{\mathrm{i}} + A_{\mathrm{e}})\mathbf{u} + A_{\mathrm{i}}\mathbf{v} = \mathbf{0}. \tag{6.2}$$

By applying a semi-implicit scheme to (6.2), we obtain the system

$$\begin{aligned} &\mathscr{B}\boldsymbol{\xi}^{k+1} = \mathbf{b} \quad \text{with } \mathscr{B} = \begin{bmatrix} C_t + \theta A_{\mathrm{i}} & \theta A_{\mathrm{i}} \\ \theta A_{\mathrm{i}} & \theta(A_{\mathrm{i}} + A_{\mathrm{e}}) \end{bmatrix}, \\ &\mathbf{b} = [C_t\mathbf{v}^k - \mathbf{i}(\mathbf{v}^k) + \mathbf{i}_a + \beta A_{\mathrm{i}}/2(\mathbf{v}^k + \mathbf{u}^k); \mathbf{0}], \quad \mathbf{v}^k = \bar{\mathbf{u}}^k - \mathbf{u}^k, \quad \boldsymbol{\xi}^{k+1} = [\mathbf{v}^{k+1}; \mathbf{u}^{k+1}]. \end{aligned} \tag{6.3}$$

The parameters θ and β depend on the used scheme; we refer to Table 1 for typical values of θ. In the following we shall consider the case $\theta = 1$ and $\beta = 0$. The symmetric matrix $\mathscr{B}$ is positive semidefinite. Indeed, let $\mathbf{0} \neq \boldsymbol{\xi} = [\mathbf{v}; \mathbf{u}] \in \mathbb{R}^{2n}$, then

$$\begin{aligned} \boldsymbol{\xi}^{\mathrm{T}}\mathscr{B}\boldsymbol{\xi} &= \mathbf{v}^{\mathrm{T}}C_t\mathbf{v} + \mathbf{v}^{\mathrm{T}}A_{\mathrm{i}}\mathbf{v} + \mathbf{u}^{\mathrm{T}}(A_{\mathrm{i}} + A_{\mathrm{e}})\mathbf{u} + 2\mathbf{v}^{\mathrm{T}}A_{\mathrm{i}}\mathbf{u} \\ &= \mathbf{v}^{\mathrm{T}}C_t\mathbf{v} + (\mathbf{v}+\mathbf{u})^{\mathrm{T}}A_{\mathrm{i}}(\mathbf{v}+\mathbf{u}) + \mathbf{u}^{\mathrm{T}}A_{\mathrm{e}}\mathbf{u} \geqslant 0 \end{aligned}$$

since A_s, $s = \mathrm{i}, \mathrm{e}$ are positive semidefinite and C_t is positive definite. Moreover, $\mathscr{B}[\mathbf{0}; \mathbf{e}] = 0$. Note that $\mathscr{B}$ is denser then $\mathscr{A}$ in the $(\bar{u}, u)$ formulation.

Classically, the system with $\mathscr{B}$ is solved by means of a nested iteration [21,30,20] that can be explicitly stated as a block Gauss–Seidel method involving the two diagonal blocks [9]. In the next section, we discuss the solution of the iterative method associated with a predictor–corrector scheme; other schemes could also be considered for comparison purposes.

6.1. Block Gauss–Seidel method

When a predictor–corrector scheme of second order is used for the time discretization, the predictor step requires the solution of two successive linear systems of dimension n with coefficient matrices $C_t + A_{\mathrm{i}}/2$ and $A_{\mathrm{i}} + A_{\mathrm{e}}$. In the corrector step, the following linear system of dimension $2n$ need be solved [9]:

$$\mathscr{M}\boldsymbol{\xi} = \mathbf{b} \quad \text{with } \mathscr{M} = \begin{bmatrix} C_t + \dfrac{A_{\mathrm{i}}}{2} & \dfrac{A_{\mathrm{i}}}{2} \\ A_{\mathrm{i}} & A_{\mathrm{i}} + A_{\mathrm{e}} \end{bmatrix} \tag{6.4}$$

with $\boldsymbol{\xi}^{k+1} = [\mathbf{v}^{k+1}; \mathbf{u}^{k+1}]$ and $\mathbf{b}$ properly defined. Note that, modulo a scaling of the second block row, a symmetric system can be obtained that corresponds to system (6.3) with $\theta = 1/2$.

In [9], system (6.4) was solved by means of a block Gauss–Seidel method as follows:

Given $\mathbf{v}^0, \mathbf{u}^0$ for $r = 0, 1, \ldots, v$

$$\begin{aligned} &\left(C_t + \frac{A_{\mathrm{i}}}{2}\right)\mathbf{v}^{r+1} = -\frac{1}{2}A_{\mathrm{i}}\mathbf{u}^r + \mathbf{b}_1, \\ &(A_{\mathrm{i}} + A_{\mathrm{e}})\mathbf{u}^{r+1} = -A_{\mathrm{i}}\mathbf{v}^{r+1}, \end{aligned} \tag{6.5}$$

where $\mathbf{b}_1$ is a properly determined vector. As in the predictor step, in the corrector phase two large linear systems with $C_t + A_{\mathrm{i}}/2$ and $A_{\mathrm{i}} + A_{\mathrm{e}}$ need be solved at each iteration r. This is done by means of

the preconditioned conjugate gradient method. While solving with $C_t+A_i/2$ is very efficient, due to its diagonal dominance, the iterative solution of the system with the semidefinite matrix A_i+A_e may be extremely slow in practice, making the entire procedure very expensive. SSOR preconditioning was used in [9] in both the definite and semidefinite cases. Experiments with other preconditioners such as incomplete factorizations with fill-in and threshold did not lead to significantly better performance. Nevertheless, the implemented block Gauss–Seidel iteration converges in very few iterations (cf. the experiments section) and the method does not seem to suffer from the approximate solution of the inner systems. The efficiency of the block Gauss–Seidel iteration when the inner linear systems are solved exactly can be explained as follows. The splitting used in (6.5) is

$$\mathscr{M} = \begin{bmatrix} C_t + \dfrac{A_i}{2} & 0 \\ A_i & A_i + A_e \end{bmatrix} - \begin{bmatrix} 0 & -\dfrac{A_i}{2} \\ 0 & 0 \end{bmatrix}.$$

Therefore, the splitting is of the form $\mathscr{M} = \mathscr{H} - \mathscr{N}$ with both $\mathscr{M}$ and $\mathscr{H}$ singular. It is known that in this case the Gauss–Seidel iteration is convergent if and only if $\varrho(\mathscr{H}^\dagger \mathscr{N}) < 1$, where $\mathscr{H}^\dagger$ stands for the Moore–Penrose inverse of $\mathscr{A}$ [6]. The consistency of the problem allows us to explicitly write the iteration matrix as

$$B_{GS} = \mathscr{H}^\dagger \mathscr{N} = \begin{bmatrix} 0 & -(C_t + \frac{A_i}{2})^{-1} \frac{A_i}{2} \\ 0 & (A_i + A_e)^\dagger A_i (C_t + \frac{A_i}{2})^{-1} \frac{A_i}{2} \end{bmatrix}. \tag{6.6}$$

In the following we give bounds for the spectral radius of the iteration matrix $\mathscr{H}^\dagger \mathscr{N}$ in terms of the largest eigenvalue of the pencil (A_i, C_t). The proof is postponed to the appendix. We recall that the spectral radius $\varrho(X)$ of a square matrix X is given by its largest eigenvalue in modulo.

Proposition 6.1. *The iteration matrix B_{GS} defined in* (6.6) *satisfies*

$$\varrho(B_{GS}) \leqslant \frac{\lambda_{\max}(A_i, C_t)}{2 + \lambda_{\max}(A_i, C_t)},$$

where $\lambda_{\max}(A_i, C_t)$ is the largest eigenvalue of (A_i, C_t).

The largest eigenvalue of (A_i, C_t) is not much larger than one, therefore, the bound of Proposition 6.1 implies that $\varrho(B_{GS}) \ll 1$, so that good convergence of the block Gauss–Seidel iteration is expected.

As an alternative to the block Gauss–Seidel procedure and in order to compare the efficiency of the different formulations, we also propose to solve system (6.3) using the conjugate gradient method preconditioned by block SSOR, as done for (3.7). Due to the denser block structure of the coefficient matrix, different performance is expected than in the case of system (4.1).

7. Computational experiments

The computational experiments were carried out to compare the different formulations and solvers introduced in the previous sections. We deal with a parallelepipedal domain Ω representing a block of myocardium of dimension $(l_1, l_2, l_3) = 1.5 \times 1.5 \times 0.3$ cm. For simplicity, but without loss of generality, we deal with the case of no fiber rotation, i.e., the vector $\mathbf{a}(\mathbf{x})$ in (2.2), locally tangent

Table 2

$\chi = 1000\ \mathrm{cm}^{-1}$	$G = 4 \times 10^{-4}\ \Omega^{-1}\ \mathrm{cm}^{-2}$	$c_m = 0.8\ \mu\mathrm{F}\ \mathrm{cm}^{-2}$
$i_{\mathrm{app}} = 0.8\ \mathrm{A}\ \mathrm{cm}^{-3}$	$v_{\mathrm{p}} = 100\ \mathrm{mV}$	$v_{\mathrm{th}} = 10\ \mathrm{mV}$
$\sigma_l^{\mathrm{e}} = 2.5 \times 10^{-3}$,	$\sigma_t^{\mathrm{e}} = 1.25 \times 10^{-3}$	$\Omega^{-1}\ \mathrm{cm}^{-1}$
$\sigma_l^{\mathrm{i}} = 2 \times 10^{-3}$,	$\sigma_t^{\mathrm{i}} = 4.16 \times 10^{-4}$	$\Omega^{-1}\ \mathrm{cm}^{-1}$

to the fibers, is assumed constant. We consider a uniform mesh on Ω made up of parallelepipedal elements with edges $(h_1, h_2, h_3) = (l_1/60, l_2/60, l_3/18)$, yielding $n = 70\,699$ nodes in Ω hence a system $\mathscr{A}$ of size 141 398. The space step in the x, y directions is 2.5×10^{-2} cm, whereas the time step τ was chosen equal to 4×10^{-2} ms. With these time and space steps we can obtain stable and accurate results as shown by the validation carried out in [9].

The propagation was elicited by applying a current pulse of $0.8\ \mathrm{A/cm}^3$ lasting 0.5 ms; hence in (3.6) a value of $I_{\mathrm{app}} = 0.8$ is applied to each grid node of the stimulated region. We considered the same parameter calibration used in [9] as reported in Table 2.

All experiments correspond to a typical temporal instant in the time step evolution, so that the right-hand side includes information generated during the previous time steps.

The following methods are compared

On system (6.3):

- Block Gauss–Seidel with inner preconditioned CG:
  - Incomplete factorization preconditioner: ICT$(0, 10^{-8})$
  - SSOR(ω) preconditioner

On system (4.1) and system (6.3)

- Preconditioned CG:
  - SSOR(ω) preconditioner on original matrix.
  - Block SSOR(ω) preconditioner on permuted matrix.
  - Incomplete factorization preconditioner on permuted matrix: ICT$(0, 10^{-8})$.

All numerical tests were done in fortran on a Sun Enterprise 4500, 400 MHz, 2GBytes RAM. Comparision between different methods is carried out by measuring elapsed time (fortran function `dtime`). In our large scale application, memory requirements also represents an important comparison reference so that methods employing comparable memory allocations are considered.

The incomplete factorization code we used in our experiments is the symmetric ICT algorithm from the ICT package by Chow and Saad [8]. The algorithm performs an incomplete Cholesky factorization of the given matrix [33], allowing a number of additional nonzero elements per row corresponding to a chosen fill-in parameter, while dropping entries that are below a tolerance chosen by the user. In the experiments reported in the tables, we have used dropping tolerance equal to 10^{-8}. For comparison purposes, no fill-in was allowed, although it is well known that incomplete factorization preconditioners may be very effective if fill-in is allowed. For instance, on our problem, with fill-in equal to 5 and 10, the total elapsed time goes down to 17 s in both cases, with 35 and 29 iterations, respectively. However, memory requirements become prohibitive for our 3D problem, since 1.73/2.01 and 2.44/2.72 (real/integer) $\times 10^6$ memory allocations are needed, respectively, to store the denser preconditioners.

Table 3
Performance of block Gauss–Seidel method on (6.3). Different preconditioning strategies for the inner system solution with coefficient matrix $A_i + A_e$

Inner preconditioner	Elapsed time
SSOR $\omega = 1$	137.9
SSOR $\omega = 1.7$	65.1
ICT lfil $= 0$, droptol $= 10^{-8}$	123.6

Table 4
Performance of CG on formulation $(\bar{u}, u)$ in (3.7) with different preconditioning strategies

Preconditioner	Its.	Elapsed time
SSOR $\omega = 1$	75	34.5
SSOR $\omega = 1.7$	59	26.0
ICT lfil $= 0$, droptol $= 10^{-8}$	59	22.2
Block SSOR $\omega = 1$	58	22.8
Block SSOR $\omega = 1.7$	33	13.0

All iterative solvers were stopped when the relative residual norm of the original linear system was less than 10^{-6}. The block Gauss–Seidel method on the standard (v, u) formulation entails the solution of two symmetric systems at each iteration, as shown in (6.5), by means of the CG method. The coefficient matrix of the first system is diagonally dominant and few iterations (about 6) are needed to reach a relative residual norm of 10^{-6}, while the solution of the linear system with $A_i + A_e$ requires preconditioning. Different preconditioning techniques were explored and the relevant timing results are reported in Table 3; since the coefficient matrices do not change with the outer iteration, the preconditioner was built at the beginning of the process and kept throughout the entire cycle. The number of iterations to converge for the block Gauss–Seidel method is very low: typically less than 10 iterations are required. Nevertheless, the inner linear system solution is very expensive and penalizes the overall performance, when measured in terms of computer time. This drawback is peculiar of inner–outer iterations, where the convergence of the inner method may dramatically influence the total computational cost.

We note that SSOR with optimal parameter is very efficient, also with respect to incomplete factorization preconditioning with no fill-in.

Tables 4 and 5 instead refer to numerical experiments with preconditioned CG on the full matrix with formulations $(\bar{u}, u)$ and (v, u), respectively. In both tables, the first column reports the preconditioner used, while the second and third columns report the number of iterations and elapsed time, respectively. It immediately appears that most elapsed timings are much lower than those of the block Gauss–Seidel approach in Table 3, showing that dealing with the whole block system is very convenient. We also notice that SSOR preconditioning, both in its scalar and block form, is very sensitive to the choice of the parameter ω. Moreover, the improvement of the block version of SSOR over its scalar counterpart can be clearly appreciated, especially in the $(\bar{u}, u)$ formulation.

Table 5
Performance of CG on formulation (v,u) in (6.3) with different preconditioning strategies

Preconditioner	Its.	Elapsed time
SSOR $\omega = 1$	66	42.3
SSOR $\omega = 1.7$	36	23.4
ICT lfil $= 0$, droptol $= 10^{-8}$	47	25.2
Block SSOR $\omega = 1$	54	27.7
Block SSOR $\omega = 1.7$	32	18.0

Table 6
Performance for a run of 925 time steps. In the block SSOR and SSOR preconditioners is $\omega = 1.7$

Formulation	Algebraic method	Elapsed time
$(\bar{u},u)$	Block SSOR preconditioned CG	3 h 20 min
(v,u)	Block SSOR preconditioned CG	4 h 37 min
(v,u)	Gauss–Seidel + inner SSOR preconditioned CG	16 h 42 min

Comparing Tables 4 and 5, we notice that for $\omega = 1.7$, block SSOR takes roughly the same number of iterations to converge in the two formulations. However, the gap between the elapsed timings is larger than expected. This is clearly due to the different cost per iteration of the two preconditioned schemes, caused by the higher density of the coefficient matrix in the (v,u) formulation.

8. Conclusions

From our numerical experiments, we come to the conclusion that dealing with the full matrix is indeed advantageous with respect to using a nested procedure, confuting the common belief in the field that considering smaller systems will always lead to less expensive schemes. It is also very interesting that the $(\bar{u},u)$ formulation, commonly only used for the theoretical analysis, is in fact superior to the (v,u) formulation in a numerical performance context.

We observe a factor of five reduction in elapsed time when using the block SSOR preconditioned CG on the $(\bar{u},u)$ formulation, with respect to the usual nested method on the (v,u) formulation. The high performance of this approach can be further appreciated when considering an entire evolution run, which consists of several hundred time steps. More specifically, we know that the block of myocardium chosen is completely excited in about 37 ms after central face stimulation. The approximate total time required to simulate the whole process using a time step $\tau = 4 \times 10^{-2}$ ms, hence considering 925 time steps, is reported in Table 6.

The above results clearly indicate the efficiency of the new algebraic method and show the better computational performance of the $(\bar{u},u)$ formulation. Therefore these results can be considered as a very promising step towards 3-D large scale simulations of the whole myocardial excitation process.

By reordering the blocks in (4.1), the system can be written as

$$\begin{bmatrix} -C_t & C_t + A_\mathrm{i} \\ C_t + A_\mathrm{e} & -C_t \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}.$$

The reordered matrix has block diagonal matrices that are diagonal, so that the matrix structure is reminiscent of that obtained with a red-black mesh ordering [33], for which a Schur complement method is known to be effective. In our case this approach did not seem to perform well. Nevertheless, future efforts will be devoted to further exploit the block structure of the coefficient matrix in the system in order to enhance the preconditioning procedure.

Acknowledgements

We thank P. Colli Franzone and L. Guerri for helpful discussions and Y. Notay for making his paper [27] available to us.

Appendix

In this appendix we prove the bounds stated in Propositions 4.2 and 6.1.

Proof of Proposition 4.2. By explicitly writing the iteration matrix B_{GS}, we see that if λ is an eigenvalue of B_{GS} then either $\lambda = 0$ or λ is an eigenvalue of the nonsingular matrix $(C_t+A_\mathrm{e})^{-1}C_t(C_t+A_\mathrm{i})^{-1}C_t$. The eigenvalue problem $(C_t + A_\mathrm{e})^{-1}C_t(C_t + A_\mathrm{i})^{-1}C_t\mathbf{x} = \lambda\mathbf{x}$ can be written as

$$C_t(C_t + A_\mathrm{i})^{-1}C_t\mathbf{x} = \lambda(C_t + A_\mathrm{e})\mathbf{x}. \tag{A.1}$$

Since C_t is positive definite then $\lambda > 0$ and the eigenvalue $\lambda = 1$ is associated with the eigenvector $\mathbf{x} = \mathbf{e}$. Moreover, we can write

$$(I + \tilde{A}_\mathrm{i})(I + \tilde{A}_\mathrm{e})\mathbf{z} = \frac{1}{\lambda}\mathbf{z} \quad \mathbf{z} = C_t^{1/2}\mathbf{x},$$

where $\tilde{A}_s = C_t^{-1/2}A_sC_t^{-1/2}$, $s = \mathrm{i}, \mathrm{e}$. Therefore, for $\mathbf{z} \perp C_t^{1/2}\mathbf{e}$ we have

$$||\mathbf{z}||^2 + \lambda_{\min}(\tilde{A}_\mathrm{e})||\mathbf{z}||^2 \leqslant \mathbf{z}^{\mathrm{T}}(I + \tilde{A}_\mathrm{e})\mathbf{z} = \frac{1}{\lambda}\mathbf{z}^{\mathrm{T}}(I + \tilde{A}_\mathrm{i})^{-1}\mathbf{z} \leqslant \frac{1}{\lambda}\frac{1}{1 + \lambda_{\min}(\tilde{A}_\mathrm{i})}||\mathbf{z}||^2,$$

where $\lambda_{\min}(\cdot)$ indicates the smallest nonzero eigenvalue of the given matrix, from which

$$\lambda \leqslant \frac{1}{(1 + \lambda_{\min}(\tilde{A}_\mathrm{e}))(1 + \lambda_{\min}(\tilde{A}_\mathrm{i}))} < 1. \qquad \square$$

Proof of Proposition 6.1. The eigenvalues of B_{GS} in (6.6) are either zero or are real eigenvalues of the pencil $(A_\mathrm{i}(C_t + A_\mathrm{i}/2)^{-1}A_\mathrm{i}/2, A_\mathrm{i} + A_\mathrm{e})$. Note that the associated generalized eigenvalue problem is

singular, since

$$\frac{A_\mathrm{i}}{2}\left(C_t+\frac{A_\mathrm{i}}{2}\right)^{-1}A_\mathrm{i}\mathbf{x}=\lambda(A_\mathrm{i}+A_\mathrm{e})\mathbf{x} \tag{A.2}$$

is satisfied for $\mathbf{x}=\mathbf{e}$ and for any λ. Nonetheless, since the system $(A_\mathrm{i}+A_\mathrm{e})Y=A_\mathrm{i}/2(C_t+A_\mathrm{i}/2)^{-1}A_\mathrm{i}$ is consistent, we can restrict to the orthogonal complement of $\mathbf{e}$, where the eigenvalue problem is regular. For $\mathbf{x}\perp\mathbf{e}$, let us write

$$\frac{1}{2}\mathbf{x}^\mathrm{T}A_\mathrm{i}\left(C_t+\frac{A_\mathrm{i}}{2}\right)^{-1}A_\mathrm{i}\mathbf{x}=\lambda\mathbf{x}^\mathrm{T}A_\mathrm{i}\mathbf{x}+\lambda\mathbf{x}^\mathrm{T}A_\mathrm{e}\mathbf{x}. \tag{A.3}$$

Let θ be a real nonzero eigenvalue, solution of the eigenvalue problem

$$A_\mathrm{i}(2C_t+A_\mathrm{i})^{-1}A_\mathrm{i}\mathbf{x}=\theta A_\mathrm{i}\mathbf{x}\quad\mathbf{x}\perp\mathbf{e}. \tag{A.4}$$

Let $A_\mathrm{i}=GG^\mathrm{T}$ with $G\,n\times(n-1)$ full rank matrix, $\mathbf{e}\perp\mathrm{span}\{G\}$. Then (A.4) can be rewritten as $GG^\mathrm{T}(2C_t+GG^\mathrm{T})^{-1}GG^\mathrm{T}\mathbf{x}=\theta GG^\mathrm{T}\mathbf{x}$. Using the Sherman–Morrison formula (see e.g. [15]), we have $G^\mathrm{T}(2C_t+GG^\mathrm{T})^{-1}G=G^\mathrm{T}(2C_t)^{-1}G(I+G^\mathrm{T}(2C_t)^{-1}G)^{-1}$. Simple manipulations give

$$GG^\mathrm{T}(2C_t)^{-1}G\mathbf{z}=\theta G(I+G^\mathrm{T}(2C_t)^{-1}G)\mathbf{z},\quad\mathbf{z}=(I+G^\mathrm{T}(2C_t)^{-1}G)^{-1}G^\mathrm{T}\mathbf{x},$$

from which $(1-\theta)GG^\mathrm{T}C_t^{-1}G\mathbf{z}=2\theta G\mathbf{z}$. Since it must be $\theta\neq1$, then we can write the eigenvalue problem as

$$A_\mathrm{i}C_t^{-1}\mathbf{y}=\frac{2\theta}{1-\theta}\mathbf{y},\quad\mathbf{y}=G\mathbf{z},$$

that is, $\lambda=2\theta/(1-\theta)$ is a nonzero eigenvalue of the pencil (A_i,C_t). Therefore,

$$\theta=\frac{\lambda}{2+\lambda}\leqslant\frac{\lambda_{\max}(A_\mathrm{i}C_t)}{2+\lambda_{\max}(A_\mathrm{i},C_t)}. \tag{A.5}$$

Noticing that $\mathbf{x}^\mathrm{T}A_\mathrm{e}\mathbf{x}>0$ for $\mathbf{x}\perp\mathbf{e}$, from (A.3) we have

$$\frac{1}{2}\mathbf{x}^\mathrm{T}A_\mathrm{i}\left(C_t+\frac{A_\mathrm{i}}{2}\right)^{-1}A_\mathrm{i}\mathbf{x}>\lambda\mathbf{x}^\mathrm{T}A_\mathrm{i}\mathbf{x}$$

and using the bound (A.5) we obtain

$$\lambda\leqslant\frac{\lambda_{\max}(A_\mathrm{i},C_t)}{2+\lambda_{\max}(A_\mathrm{i},C_t)}$$

and the bound is proved. □

References

[1] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit–explicit methods for time-dependent, PDE's SIAM J. Numer. Anal. 32 (1995) 797–823.
[2] C.C. Ashcraft, R.G. Grimes, On vectorizing incomplete factorization and SSOR preconditioners, SIAM J. Sci. Comput. 9 (1988) 122–151.
[3] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1996.
[4] O. Axelsson, V.A. Barker, Finite Element Solution of Boundary Value Problems — Theory and Computation, Computer Science and Applied Mathematics, Academic Press, London, 1984.

[5] Z. Bai, A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations, Adv. Comput. Math. 10 (1999) 169–186.
[6] A. Berman, R.J. Plemmons, Cones and iterative methods for best least squares solutions of linear systems, SIAM J. Numer. Anal. 11 (1974) 145–154.
[7] A. Berman, R.J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1994, xx+340.
[8] E. Chow, Y. Saad, ICT package, 1999, private communication.
[9] P. Colli Franzone, L. Guerri, Spreading of excitation in 3-D models of the anisotropic cardiac tissue. I: validation of the eikonal model, Math. Biosci. 113 (1993) 145–209.
[10] P. Colli Franzone, L. Guerri, M. Pennacchio, B. Taccardi, Spread of excitation in 3-D models of the anisotropic cardiac tissue. II: effects of fiber architecture and ventricular geometry, Math. Biosci. 147 (1998) 131–171.
[11] P. Colli Franzone, L. Guerri, M. Pennacchio, B. Taccardi, Spread of excitation in 3-D models of the anisotropic cardiac tissue. III: effects of ventricular geometry and fiber structure on the potential distribution, Math. Biosci. 151 (1998) 51–98.
[12] P. Colli Franzone, G. Savaré, Degenerate evolution systems modeling the cardiac electric field at micro and macroscopic level, Publ. IAN-CNR No. 1007, Pavia, 1996.
[13] A. Dax, The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations, SIAM Rev. 32 (1990) 611–635.
[14] S. Eisenstat, Efficient implementation of a class of conjugate gradient methods, SIAM J. Sci. Statist. Comput. 2 (1988) 1–4.
[15] G. Golub, C.F. Van Loan, Matrix Computations, 3rd Edition, The Johns Hopkins University Press, Baltimore, 1996.
[16] F.A. Gruzinov, L.Y. Kolotilina, A.Y. Yeremin, Block SSOR preconditionings for high-order 3D FE systems. III: incomplete BSSOR preconditionings based on p-partitionings, Numer. Linear Algebra Appl. 4 (1997) 393–423.
[17] D.L. Harrar, Analytically and implementationally optimal 2-color SSOR preconditioning on vector and parallel supercomputers, in: J.G. Lewis (Ed.), Applied Linear Algebra, Proceedings of the Fifth SIAM Conference, Snowbird, SIAM, Philadelphia, PA, June 1994, pp. 556–561.
[18] D.L. Harrar, J.M. Ortega, Optimum m-step SSOR preconditioning, J. Comput. Appl. Math. 24 (1988) 195–198.
[19] C.S. Henriquez, Simulating the electrical behavior of cardiac tissue using the bidomain model, Crit. Rev. Biomed. Eng. 21 (1993) 1–77.
[20] C.S. Henriquez, A.L. Muzikant, C.K. Smoak, Anisotropy, fiber curvature, and bath loading effects on activation in thin and thick cardiac tissue preparations: simulations in a three-dimensional bidomain model, J. Cardiovasc. Electrophysiol. 7 (5) (1996) 424–444.
[21] N.F. Hooke, Efficient simulation of action potential propagation in a bidomain, Ph.D. Thesis, Duke University, 1992.
[22] N. Hooke, C.S. Henriquez, P. Lanzkron, D. Rose, Linear algebraic transformations of the bidomain equations: implications for numerical methods, Math. Biosci. 120 (1994) 127–145.
[23] J.P. Keener, An eikonal-curvature equation for action potential propagation in myocardium, J. Math. Biol. 29 (1991) 629–651.
[24] J.P. Keener, K. Bogar, An numerical method for the solution of the bidomain equations in cardiac tissue, Chaos 8 (1) (1998) 234–241.
[25] The MathWorks, Inc., MATLAB User's Guide, MathWorks, Natick, MA. 01760, Jan 1998.
[26] G. Meurant, Computer Solution of Large Linear Systems, Studies in Mathematics and its Applications, Elsevier Science, The Netherlands, 1999.
[27] Y. Notay, Polynomial acceleration of iterative schemes associated with subproper splittings, J. Comput. Appl. Math. 24 (1988) 153–167.
[28] Y. Notay, Incomplete factorizations of singular linear systems, BIT 29 (1989) 682–702.
[29] Y. Notay, Solving positive (semi) definite linear systems by preconditioned iterative methods, in: O. Axelsson, L. Kolotilina (Eds.), Preconditioned Conjugate Gradient Methods, Lectures Notes in Mathematics, Vol. 1457, Springer, Berlin, 1990, pp. 105–125.
[30] A. Pollard, N. Hooke, C. Henriquez, Cardiac propagation simulation, in: T. Pilkington, B. Loftis, J.F. Thompson, S. Woo, T. Palmer, T. Budinger (Eds.), High Performance Computing in Biomedical Research, 1992, pp. 319–358.
[31] W. Quan, S.T. Evans, H.M. Hastings, Efficient integration of a realistic two-dimensional cardiac tissue model by domain decomposition, IEEE Trans. Biomed. Eng. 45 (3) (1998) 372–385.

[32] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer, Berlin, 1994.
[33] Y. Saad, Iterative Methods for Sparse Linear Systems, The PWS Publishing Company, 1996.
[34] H.I. Saleheen, K.T. Ng, A new three-dimensional finite difference bidomain formulation for the inhomogeneous anisotropic cardiac tissues, IEEE Trans. Biomed. Eng. 45 (1) (1998) 15–25.
[35] E.J. Vigmond, J. Leon, Computationally efficient model for simulating electrical activity in cardiac tissue with fiber rotation, Ann. Biomed. Eng. 27 (1999) 160–170.
[36] T. Washio, K. Hayami, Parallel block preconditioning based on SSOR and MILU, Numer. Linear Algebra Appl. 6 (1994) 533–553.